

37.5

Students Name:

ID:

Section:

1. Briefly explain with an example what is meant by each of the following terms? (6 points)

- a. Method
- b. Attribute
- c. Encapsulation

a. Method is an implementation of an operation

```
void test()
```

```
{
```

```
    ; ← Method
```

```
}
```



b. Attribute is an entity represents conceptual idea

```
class E
```

```
    private test ← Attribute
```

```
    public void test();
```

```
}
```



c. Encapsulation

Ability to define attributes and methods within the same unit

```
class Enc
```

```
    private int i;
```

```
    private int b;
```

```
    public void set()
```

```
    {
```

```
        i=1;
```

```
        b=2;
```

```
    }
```

```
    public Add void add()
```

```
    {
```

→ Encapsulation



Students Name:

ID:

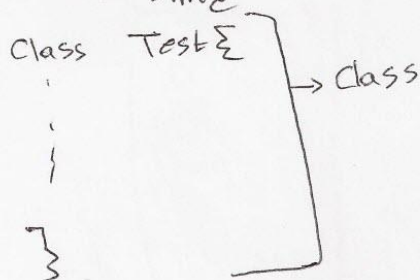
Section:

2. Explain the difference between each of the following concepts; illustrate your answer with at least an example. (10 points)

- a. Object and Class
- b. Aggregation and Composition

Object is an attribute represent a physical or a logical concept and

while class is ~~some~~ a concept consists of objects and operations at the same time



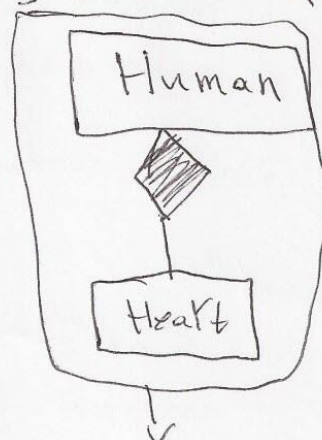
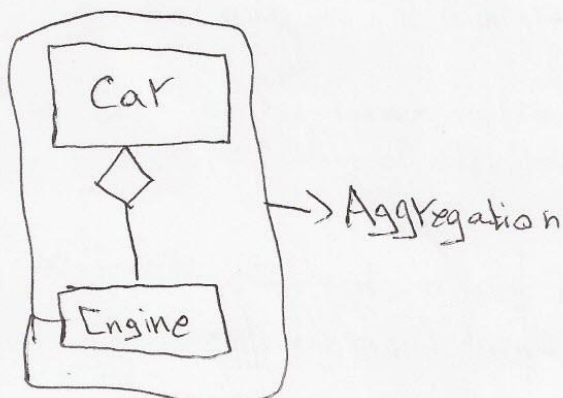
Test variable;

↓  
object

Aggregation is an entity consists of several parts, if we delete the whole ~~part~~ entity, parts still exists.

~~Aggregation is an entity consists of several parts, if one of these parts are affected it will not affect the whole entity.~~

~~Some~~ while composition is an entity consists of several parts, if we delete the whole entity then the parts are also deleted





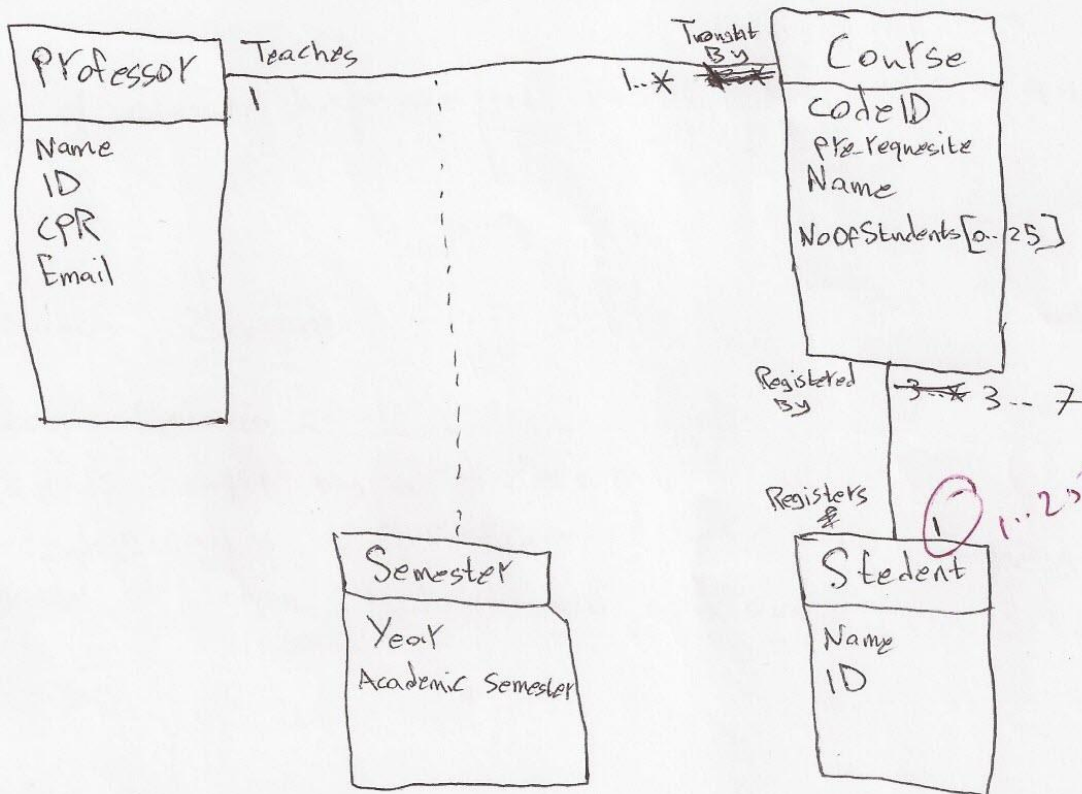
University of Bahrain  
College of IT  
Department of Computer Science  
2<sup>nd</sup> 2013-2014  
ITCS341 – Object-Oriented Concepts  
First Test

Students Name:

ID:

Section:

3. In a university registration system a professor is assigned to one or more courses to be offered in the current semester. An offered course will have up to 25 students registered for it. Analyze this situation based on your understanding and design a suitable class model for the classes and their relationships for the system. Your answer should include multiplicity and end names. State your reasons for the classes, relationship(s), multiplicity, and end names. (14 points)



### Multiplicity Reasons

- \* One professor might teach 1 or more courses, the reason of this multiplicity is the question. It is mentioned in the question.
- \* Each course have up to 25 students, in this case we can use multiplicity of attributes.
- \* Each student might register 3 to 7 courses, it is an assumption according to UOB system.

Students Name:

ID:

Section:

Reason of association classes

Semester is an association class because we need to keep track of the semester's professors used to teach in, and which courses.

Reason of classes because it is stated in the question

End Names Reasons

- Professor teaches course
- Course is taught by a professor

Because logically it is represented like this, same goes to the end name between student and courses.



University of Bahrain  
College of IT  
Department of Computer Science  
2<sup>nd</sup> 2013-2014  
ITCS341 – Object-Oriented Concepts  
First Test

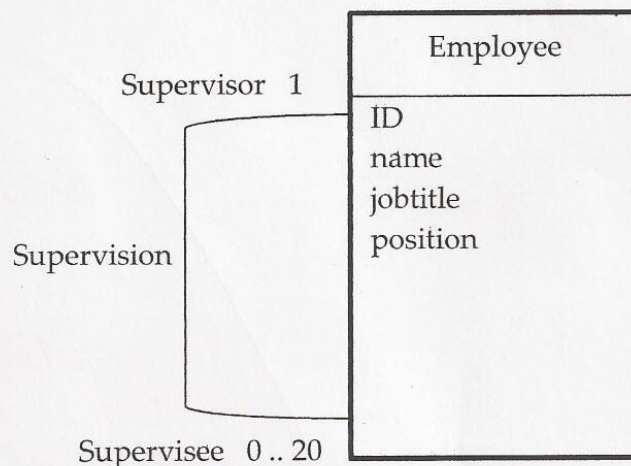
Students Name:

ID:

Section:

4. In a Human Resources system an employee has a unique ID, name, job-title, position, and may supervise up-to 20 other employees (i.e. an employee reports to another employee - his supervisor). The class diagram is shown below. You are required to write the necessary Java code to implement the class. Including all its attributes and those required to implement the relationship. Explain your choice of data structure for the relationship. (10 points)

No methods need to be specified or coded.



```
class Employee {
```

```
    private int ID;
```

```
    private String Name;
```

```
    private String JobTitle;
```

```
    private String Position;
```

```
    private int Supervise[];
```

```
    private Employee Supervise[];
```

```
    private Employee Supervisor;
```

```
}
```

~~\* We record the ID of the supervised employee~~

\* We need to record the details of supervised employee.

\* We need to record the details of the supervisor of